A semi-supervised approach to message stance classification

Article

Accepted Version

A semi-supervised approach to message stance classification

Georgios Giasemidis, Nikolaos Kaplis, Ioannis Agrafiotis, Jason R. C. Nurse

Abstract—Social media communications are becoming increasingly prevalent; some useful, some false, whether unwittingly or maliciously. An increasing number of rumours daily flood the social networks. Determining their veracity in an autonomous way is a very active and challenging field of research, with a variety of methods proposed. However, most of the models rely on determining the constituent messages' stance towards the rumour, a feature known as the "wisdom of the crowd". Although several supervised machine-learning approaches have been proposed to tackle the message stance classification problem, these have numerous shortcomings. In this paper we argue that semi-supervised learning is more effective than supervised models and use two graph-based methods to demonstrate it. This is not only in terms of classification accuracy, but equally important, in terms of speed and scalability. We use the Label Propagation and Label Spreading algorithms and run experiments on a dataset of 72 rumours and hundreds of thousands messages collected from Twitter. We compare our results on two available datasets to the state-of-the-art to demonstrate our algorithms' performance regarding accuracy, speed and scalability for real-time applications.

Index Terms—message stance, Twitter, rumours, semi-supervised, label propagation, label spreading

1 Introduction

Online content is at the centre of today's information world. A primary source of this content is social media, with the public acting as a major contributor on everything from election discussions to reports on ongoing crisis events. This level of free engagement has several benefits. For instance, it can encourage healthy discourse on pertinent topics of public interest, or it can be invaluable at supporting official responders reacting to an unfolding crisis – such as Hurricane Harvey in the US [1] or the Manchester bombings in the UK [2]. On the other hand social media can be used as a tool to disrupt and harm society. Over the last few years, we have seen a spate of misinformation and fake news intended to misguide, confuse and potentially even risk people's lives [3]. This emerging reality highlights the power of social media and the need to reliably discern genuine and useful from harmful information and noise.

There has been a wide range of research in the social media domain. Of most relevance to this work is the technical effort aimed at understanding and mitigating any disruptive impacts (e.g. malicious rumour propagation). Such work can be found as early as in Castillo et al. [4] where a series of automated methods are used to analyse the credibility of information on Twitter. The major contribution of that article has been the identification of an area of research aimed at automatically estimating the validity of rumours and features of credible content, as information spreads across social media platforms. Since then, there have been a number of proposals exploring information trust, credibility and decision-making, using technical (e.g. machine learning) and user-centred (e.g. focused on perceptions and behaviours) approaches [5], [6], [7], [8], [9]. These approaches may consider individual or aggregated content (posts, messages, etc.) within rumours and use this as a basis for credibility or trust decisions.

Most of the research on social media rumours focuses on determining their veracity. Several authors have proposed different supervised systems using temporal, structural, linguistic, network and user-oriented features [10], [11], [12], [13]. However, these approaches assume that message annotation[1] is granted. Being able to annotate messages automatically is the most important step towards determining the veracity of rumours [14].

Therefore, one of the most intriguing areas of research in the domain of social media is the problem of message (i.e. post, tweet, etc.) stance classification. Here, the aim is to determine whether a particular message supports, refutes or is neutral towards a rumour; neutral stances can be further expanded to differentiate between querying or commenting messages, as highlighted in Zubiaga et al. [15]. Stance classification is essential for the modelling of veracity in a dataset.

In this paper, we aim to improve on the current state-of-the-art by proposing a semi-supervised approach to the problem of message stance classification. We use two graph-based semi-supervised algorithms with a variety of experimental settings. We demonstrate the performance of the models on two publicly available datasets.

The novel aspects of this work are twofold. First, we propose a new machine-learning approach, based on semi-supervised learning, to the problem of message stance classification. We argue that this is a more well-rounded way to tackle the problem than using supervised learning

- *G. Giasemidis and N. Kaplis are with CountingLab Ltd., Reading, UK, e-mail: {georgios, nikos}@countinglab.co.uk.*
- *I. Agrafiotis is with Department of Computer Science, University of Oxford, Oxford, UK, email: ioannis.agrafiotis@cs.ox.ac.uk*
- *J.R.C. Nurse is with the School of Computing, University of Kent, Canterbury, UK, email: j.r.c.nurse@kent.ac.uk*

1. Message annotation refers to the classification of the message stance towards the rumour.

both in terms of accuracy and, perhaps more importantly when dealing with large and diverse datasets, in terms of computational speed and scalability. We should clarify that we do not introduce a new algorithm, but we apply an existing class of algorithms to the problem for the first time. Second, we use a larger and more diverse dataset of rumours in terms of size and topics. Our dataset consists of 15 distinct events in comparison to the publicly available ones which contain either a single event or nine events, see next sections for further details. Particularly, the lack of diversity in rumours in the publicly available datasets introduces bias/over-fitting and does not facilitate transference of knowledge, forcing the need for constant re-training.

The performance of the semi-supervised models with different features and parameters are tested on data from an earlier study [10] consisting of tweets which have been manually annotated. Our work has concluded in a semi-supervised model that consists of the Label Spreading algorithm using 1,000 Brown Clusters (i.e. groups of words that are assumed to be semantically related) as features. The model's performance is enhanced by manually annotating a small portion of the tweets. To validate our model, we apply it to two datasets; the first consists of seven rumours from the UK riots in 2011 [16], achieving an 84.9% accuracy while outperforming all benchmark and random models. The second set (the PHEME dataset) consists of 23 rumours from 9 major events [17], has a higher bias and scores 75% accuracy and outperforming all other models in terms of weighted accuracy.

The remainder of this paper is structured as follows. In Section 2 we review the literature for the state-of-art techniques in message stance classification. In Section 3 we introduce the methodology and elaborate on the semi-supervised algorithms used in this study. Section 4 presents the results from the experiments we performed using different feature sets, algorithms and kernels (Section 4.1). Furthermore, we validate the methods on two independent sets of rumours, one from the London riots and the PHEME dataset, and compare the results to the literature (Section 4.2). Finally, Section 5 concludes the paper and discusses future work.

2 Related Work

The area of rumour stance classification has recently attracted the interest of the academic community. Unlike the case of rumour veracity classification where a rumour is classified as true or false, the focus of the rumour stance classification is on individual messages. More specifically, the aim is to classify messages which contribute to a rumour into four categories, namely supporting, denying, querying and commenting. It is worth mentioning that often querying and commenting classes are either omitted or merged. Thus far, most works in this area adopt supervised methods and differ mainly in the machine learning approaches used for the classification and in the set of features that are utilised in the aforementioned algorithms [15].

The first study to delve into classification of tweets was by Mendoza et al. [11], where a collection of rumours, whose veracity was identified, was further analysed manually to establish the number of tweets that were supporting or denying the rumour. The authors classified the tweets into those denying, confirming or questioning the rumour and the end goal was to understand if the distribution of these classes can be indicative of the veracity of a rumour. Their results suggested that for rumours whose veracity was true, 95% of tweets confirmed the rumour. On the contrary, when the veracity of the rumour was deemed as false only 38% of the tweets supported the rumour. Procter et al. [18] derived similar conclusions when analysing rumours during the UK riots in 2011. They focused particularly on the popularity of the users tweeting rumours, compared patterns of how false and true rumours start and evolve and identified significant differences. Extending the afforementioned works, Andrews et al. [19], narrowed their focus on how "official" accounts can help contain a false rumour and offer best social media strategies for large organisations.

Qazvinian et al. [20], were the first to automatically classify the stance of tweets. The authors opted for Bayesian classifiers and used the same feature set that they extracted to determine the veracity of rumours. They limited their approach by considering only two classes for annotating tweets (denying and confirming). In addition, they considered only long-term rumours and focused on how users' beliefs change over this long period. In a similar vein, Hamidian et al. [21], focused on features related to time, semantic content and emoticons and their approach outperformed Qazvinian et al. They extended their previous work by introducing the Tweet Latent Vector approach and by considering what they coined as "belief features", which are features that investigate the level of committed belief for each tweet [21].

In a similar vein, Mohammad et al. [22], propose a detection system able to determine a stance of a tweet for a particular target (i.e., person, institution, event etc.) by exploring correlations between stance and sentiment. Their system draws features from word and character n-grams, sentiment lexicons and word-embedded characteristics from unlabelled data. A linear-kernel SVM classifier is utilised to produce three clusters (positive, negative and not-determined stance) with very promising results (70% F-score on SemEval-2016 data). We note that training for stance is not generalised across all tweets, but is restricted per target group.

Based on Werner's et al. [23] belief tagger, Hamidian et al. [24] created a vector indicating whether a user strongly believes in the proposition; provides a non-committed comment; reflects a weak belief in the proposition; does not expressing a belief in the proposition. Lexical features were also used based on bag-of-word sets, which consist of word unigrams. The authors then explored the performance of a set of classifiers, inter alia J48 Decision Trees, Naive Bayes networks and reported that Sequential Minimal Optimization (SMO) outperforms all approaches. Similarly to previously presented work, their approach is limited to long-term rumours only.

Zeng et al. [25] focus more on semantic and linguistic characteristics and they introduce Linguistic Inquiry and Word Count (LIWC) features, as well as n-grams and part-of-speech components. Based on their experimentation and coded dataset, they are able to achieve an accuracy of over 88% in classifying rumour stances in crisis-related posts; here, random forest models result in the best performance.

Lukasik et al., [26], [27] designed a novel approach based on Gaussian Processes. They explored its effectiveness on two datasets with varying distributions of stances. The authors report results on cases where all tweets encompassing a specific rumour are used for testing and cases where the first few tweets are added to the training set. The classifier performs very well in the latter case. The novelty of this work lies in the classification of unseen rumours since this approach can annotate tweets for each rumour separately, enabling the classification of tweets for emerging rumours in the context of fast-paced, breaking news situations.

Jin et al. [28], suggest an unsupervised topic model method to detect conflicting tweets which discuss the same topic, as a first step for determining the veracity of fake news. They determine the stance of a tweet by focusing on a pair of values (topic and view point) represented by a probability distribution over a number of tweets. The topic-viewpoint pairs are then clustered into conflicting viewpoints when the distance between topics-viewpoints of the same topic exceeds a predefined threshold. Once conflicting tweets are determined, a graph of the network containing tweets which refer to the same topic is created and an effective loss function is used to solve the optimisation problem.

Zubiaga et al. [29], introduce a novel approach that considers the sequence of replies in conversation threads in Twitter. Users' replies to one another were converted to nested tree forms and tweets were analysed not only based on their individual characteristics (content, semantics etc.) but also on their position in the conversation. Two sequential classifiers namely Linear Conditional Random Fields (CRFs) and Tree-CRFs were adopted and eight datasets were used for validation with Tree-CRF performing slightly better than the Linear-CRF.

Kochkina et al. [30], proposed a deep-learning approach adopting Long Short-Term Memory networks (LSTMs) for sequential classification. They perform a pre-processing step by removing non-alphabetic characters and they tokenise the words. They further extract word vectors based on Google's *word2vec* model [31], count negation words and punctuation, identify the presence of attachments, follow the relation of content to other tweets in the discussion and count the content length. The model is trained using the categorical cross entropy loss function, however, they report that their approach is unable to distinguish any tweets denying a rumour, which are the most under-represented in their dataset. They note that these tweets are mostly misclassified as commenting and theorise that an increased amount of labelled data would improve the performance of their approach.

It is also worth mentioning some approaches that have critically reflected on the literature of stance categorisation. Shu et al. [32], present an overview of emerging research regarding fake news and stance classification. They elicit features from psychology and social theories, linguistic examination, as well as network and user characteristics and identify a number of models that can potentially utilise such features. One of these is stance-based approaches which centre around a single post and propagation-based approaches which focus on how tweets about a theme are interconnected. They conclude their survey by proposing a number of different datasets for testing of novel systems and suggest evaluation methods (i.e., Accuracy score, F-score). Finally, Ferreira and Vlachos [33] examine rumour detection in environments that are not related to social media. They present a dataset that comprises online articles and propose tailored features to the structure of the articles. They utilise logistic regression to categorise articles into those which are verified and those that are false with relative success (73% accuracy).

3 METHODOLOGY

In this work, we propose that the problem of message stance classification is more efficiently approached by semi-supervised learning algorithms. We argue that other supervised machine learning approaches, even though they may achieve marginal higher accuracy in limited datasets, they do not perform satisfactorily at large scale, which is more relevant to real-life applications. To this end, we use a class of graph-based semi-supervised algorithms, namely Label Propagation and Label Spreading, to illustrate our arguments. It is worth noting that other semi-supervised methods could be used as well, but a full comparison of such semi-supervised algorithms is beyond the scope of this study. To further motivate our proposed methodology, below we briefly discuss the pros and cons of supervised, unsupervised and semi-supervised learning approaches.

First, supervised approaches have limitations as it pertains to capturing the diversity of the messages and the stance of the same message towards two opposite rumours. A supervised approach uses a large dataset of messages for training a model. When applying this model to a new message, the supervised approach usually ignores the original claim, towards which the message takes a positive, neutral or negative position. For example, consider two rumours, the first claiming "X is true" and the second claiming "Y is true". In a supervised approach, a message saying "X is true and Y is not true" trained on the first rumour will always be classified in the "supporting" class, irrespective of whether it refers to the first or the second rumour.

There have been hybrid supervised approaches, e.g. [16], that take into account annotated messages from the rumour under consideration in order to enhance performance. However, these approaches have a serious drawback. In a live environment, where speed is as essential as accuracy, they require the retraining of a large set of annotated messages for every new rumour. The training of accurate supervised models can be computationally very expensive and time-consuming, which makes such hybrid approaches inappropriate for real-life applications.

Unsupervised machine learning splits the messages into distinct clusters, but it provides no details about the content of these clusters. it is therefore necessary to manually inspect a sample of messages from each cluster to decide whether the cluster consists of supporting, denying or neutral messages.

This brings us to the semi-supervised learning, where only a few observations are labelled and are used as seeds for the algorithm to cluster the remaining input data correctly. This approach has several advantages. First, it requires only a few labelled observations, therefore the end-user only has to manually tag a small number of messages.

Second, it is faster than supervised approaches, such as [16], which require recalibration while new messages from the rumour under consideration are being collected. Finally, it is rumour-specific, i.e. it allows the same text to be classified in different classes for different rumours, depending on the content of the rumour claim.

3.1 Data Description

Our dataset consists of the 72 rumours used in [10] [2]. The rumours were manually identified from messages (tweets) collected from Twitter, using the Twitter public API and searching for keywords related to specific events. All messages were manually annotated as *supporting*, *neutral/questioning* or *against* towards the corresponding rumour.

The size of the rumours varies from 23 to 46,807 tweets, see Figure 1a. For tweet stance classification, only the original tweets (i.e. those that are not re-tweets) must be classified as *supporting*, *neutral* or *against* towards the rumour, as re-tweets are assigned to the same class as their original tweet. Additionally, we skip non-English tweets, because the features we consider are language (here, English) specific. Figure 1b shows the distribution of the number of original English tweets for the 72 rumours.

All messages are pre-processed before feature extraction. We follow the pre-processing steps in [16]; (i) URLs, e-mails and Twitter mentions[3] are removed, (ii) text is lower-cased, (iii) all punctuations other than ",", ".", "!", "?" are removed, (iv) multiple occurrences of characters are replaced with double occurrence, and (v) extra white space is removed. Stemming, i.e. the process of reducing inflected (or sometimes derived) words to their word stem, is not performed as the Brown clusters (see Section 3.2) include whole words. Stop-words are included, because they capture important features, such as negation.

3.2 Feature Space

The messages are vectorised using three different strategies (feature sets), that are proposed in the literature (and combinations thereof).

1) 1,000 Brown clusters, denoted as "BrownC", extracted in [34] from a Twitter corpus. Every word in a tweet is placed in one of the 1,000 clusters, which represent the features.
2) Linguistic features, denoted as "Ling", such as complexity of the message, number of tentative words (e.g. "confuse", "suppose", "wonder"), that indicate uncertainty, number of swearing words, sentiment, negation, etc. These features aim to capture statistical patterns, such as tentative words being more common in messages that question a claim.
3) 2-grams to 6-grams features (abbreviated as "NGrams") of the messages in a rumour. It is worth noting that the total number of N-grams (i.e. features) varies from rumour to rumour, in contrast to the aforementioned feature sets where the features are fixed for all rumours. Another drawback is that, as new messages arrive in a live system, the feature-set is expanding. To apply this feature set, a sufficient number of messages must have been collected to capture the diversity in the N-grams.
4) A combination of Brown clusters with the sentiment and negation features from the Linguistic features ("Brown & Ling"). We choose these two linguistic features, as we would like to study the effect of the sentiment and negation in message stance classification performance. [4]

Feature selection is performed to choose the best feature set that represents the data; however we do not attempt to further reduce the size of each feature set as these are standard feature sets to represent the language in Natural Language Processing (NLP) problems [35].

3.3 Label Propagation and Label Spreading

Label propagation (LP) is a semi-supervised machine-learning method, in which observations (here messages) are represented as nodes on a graph (see [36] for a review). Consider a graph $g = (V, E)$, where $V = \{v_1, \ldots, v_n\}$ is the set of vertices (here messages), corresponding to the data (feature vectors) $X = \{\mathbf{x}_i \in \mathbb{R}^m | i = 1, \ldots, n\}$, and E is the set of edges, representing the similarities between the nodes, through a similarity matrix W. A typical choice of similarity matrix is the Gaussian kernel with width σ, i.e.

$$W_{ij} = \mathrm{e}^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2}}. \quad (1)$$

The width σ is a free parameter that requires selection. The graph could be fully connected or a k-nearest neighbours graph.

Given the graph g and a subset of labelled observations, the LP algorithm aims to propagate the labels on the graph, each node propagating its label to its neighbours until convergence.

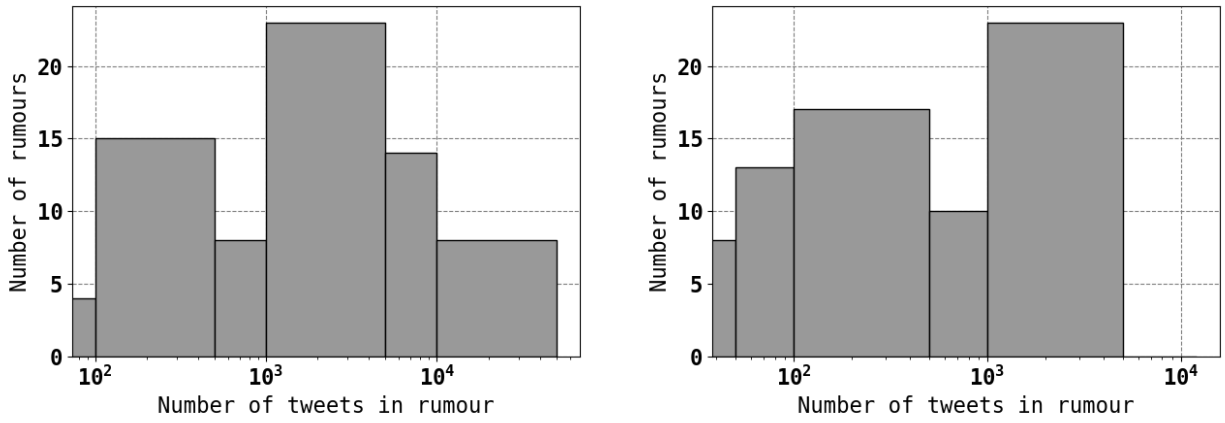
Let $\mathbf{y}_i = (y_{i,1}, y_{i,2}, y_{i,3}) \in \mathbb{R}^3$, where $y_{i,j}$ is the probability of observation i being in class $C_j \in \{-1, 0, 1\}$ representing the three classes corresponding to against, neutral and supporting messages, respectively. We denote $Y_l = \{\mathbf{y}_1, \ldots, \mathbf{y}_l\}$ the set of l labelled observations, with typically $l << n$, where $y_{i,j} = 1$ and $y_{i,k} = 0$ for $k \neq j$. Also, let $Y_u = \{\mathbf{0}, \ldots, \mathbf{0}\}$ be the set of the $n - l$ unlabelled observations, where $\mathbf{0} \in \mathbb{R}^3$ is the null vector. The algorithm proceeds as follows:

1) Compute similarity matrix W.
2) Compute the diagonal degree matrix D, $D_{ii} = \sum_j W_{ij}$.
3) Initialise the labels $\hat{Y}^{(0)} \leftarrow (Y_l, Y_u)$.
4) Iterate and impose hard-clustering: $\hat{Y}^{(t+1)} \leftarrow D^{-1}W\hat{Y}^{(t)}$ and $\hat{Y}_l^{(t+1)} \leftarrow Y_l$, where t is the iteration step, until convergence.

2. This dataset is not currently publicly available due to Intellectual Property (IP) reasons. However, the method is validated on two publicly available datasets in Section 4.2, where the results are reproducible.

3. Twitter mentions are username tags starting with the @ symbol.

4. Sentiment was extracted using the "Vader Sentiment Analyser" and negation was estimated using the "Stanford Dependency Parser", with the NLTK library in Python.



(a) Total (b) Original English only

Fig. 1: Distribution of the number of tweets in rumours: total (left) and original English only (right).

In step 4, the algorithm assigns the average label (or probability of class membership) of the neighbours of a vertex v_i to vertex v_i, i.e.

$$\hat{\mathbf{y}}_i^{(t+1)} = \frac{\sum_{j=1}^{n} W_{ij}\hat{\mathbf{y}}_j^{(t)}}{D_{ii}}, \tag{2}$$

The vertex is assigned to the class with the highest probability, i.e. $C_i = \arg\max \mathbf{y}_i$. The proof for convergence is beyond the scope of this study, but the interested reader should refer to [37] and [36, Chapter 11].

Variations of this algorithm allow for soft clustering, i.e. permitting the labelled data to change their cluster, by removing the hard-clustering assignment in step 4. This is achieved by introducing a parameter $\alpha \in [0, 1]$ in the numerator and denominator of Eq. (2) for the labelled data.

A similar algorithm, called Label Spreading (LS), uses the normalised Laplacian in the iteration step 4 above and allows the tagged observations to change classes. The algorithm becomes:

1) Compute similarity matrix W, with $W_{ii} = 0$.
2) Compute the diagonal degree matrix D, $D_{ii} = \sum_j W_{ij}$.
3) Compute the normalised graph Laplacian $L_s = D^{-1/2}WD^{-1/2}$.
4) Initialise the labels $\hat{Y}^{(0)} \leftarrow (Y_l, Y_u)$.
5) Choose a parameter $\alpha \in [0, 1]$.
6) Iterate $\hat{Y}^{(t+1)} \leftarrow \alpha L_s \hat{Y}^{(t)} + (1 - \alpha)\hat{Y}^{(0)}$ until convergence.

The algorithm has been proved to converge, see [38] and [36, Chapter 11] for further details.

The computational time of these algorithms is of order $\mathcal{O}(n^3)$ for dense graphs and $\mathcal{O}(n^2)$ for sparse ones [36, Section 11.2].

The cost function must consider both the initial labelling and the geometry of the data induced by the graph structure (i.e. edges and weights W) [36], [38],

$$\sum_{i=1}^{l} ||\hat{\mathbf{y}}_i - \mathbf{y}_i||^2 + \frac{1}{2}\sum_{i,j=1}^{n} W_{ij}||\hat{y}_i - \hat{y}_j||^2, \tag{3}$$

where the first term is a fitting constraint for the labelled data and the second term heavily penalises points that are close in the feature space but have different labels (smoothness constraint).

In this study, we use the algorithms as implemented in the *scikit-learn* library in Python [39] with a bug fix[5] that allows hard-clamping for $\alpha = 1$.

4 Experimentation and Results

In this section, we experiment with different settings of the algorithms (such as feature sets, kernels, selection of hyper-parameters), before we validate it with two publicly available datasets. The experimentation will lead to the final model and involves the following steps:

- Selection of feature set, see Section 3.2.
- Selection between Label Propagation and Label Spreading, see Section 3.3.
- Selection between Gaussian and k-nearest neighbours kernels.
- Selection of the kernel's hyper-parameter σ.

4.1 Label Propagation and Label Spreading

The Label Propagation and Label Spreading methods require the selection of a hyper-parameter, depending on the kernel used to generate the graph.

For Gaussian (or "rbf") kernel, defined in eq. (1), and fully connected graph, this is the parameter σ. We use a grid-search for finding the optimal parameter, searching in a set of values that span different orders of magnitude of σ from $\mathcal{O}(10^{-1})$ to $\mathcal{O}(10^3)$. As we see below, this range of values is sufficiently large in the search for the optimal parameter.

For k-nearest neighbours, we experiment with different numbers, k, of nearest neighbours when constructing the similarity matrix, from 5 to 50 in increments of 5.

The semi-supervised algorithm requires a sample of annotated (manually classified) messages. For our experiments

5. https://github.com/scikit-learn/scikit-learn/pull/3751/files

we annotate the first N messages (chronologically) that appear in a rumour, where the number of manually annotated messages is gradually increased $N = \{10, 20, 30, 40, 50\}$ for each rumour. Therefore, we skip rumours with less than 50 original tweets, resulting in a total of 64 rumours. We validate the performance of the model on each rumour using the messages that are not initially annotated.

We compute several performance scores, such as the accuracy, the weighted accuracy, F1-score and log-loss (entropy) scores. The accuracy is not a good performance score for biased datasets, which is the case in tweet stance classification, as most messages are in favour of the rumour. For this reason, we focus on weighted accuracy, F1-score and entropy for choosing the best-performing feature set, kernel and hyper-parameter.

We also experiment with different features sets, pre-processing steps, kernel (e.g. k-nearest neighbours ("knn")) and algorithms (e.g. Label Spreading (LS)). We summarise these results in Table 1, where we show the maximum accuracy, weighted accuracy, F1-score for $N = 50$ annotated messages. We also present the values of accuracy and F1-score at the optimal parameter ("opt param"), which is the value of the parameter (σ or k) where the weighted accuracy is maximised. The BrownC* feature set was created by altering two pre-processing steps, i.e. (i) stemming is performed, and (ii) stop-words are removed.

First, we observe that using whole words and neglecting stemming together with the use of stop-words yields better results. We investigated the effect of stemming and stop-words separately (not shown in the table however). We found that either stemming or stop-word removal results in lower performance scores. In addition, we notice that linguistic and N-gram features are poor indicators for message stance classification. Combining the Brown clusters with sentiment and negation linguistic features (BrownC & Ling) does not increase performance. This is because sentiment is also not a good indicator of message stance.

Between the available kernels, the k-nearest neighbours ("knn") appears to perform worse. We understand this to be due to the fact that the k-nearest neighbours kernel assigns either a hard-link (of unit weight) or no link between nodes with no weighting to capture the degree of similarity between messages.

Finally, the Label Spreading (LS) algorithm delivers very similar results to LP, performing marginally better. The performance plots for LP appear very similar to those in Figures 2–4, with the optimal scores being at the same regions of the σ-parameter. The two algorithms only differ, when $\alpha = 1$, at the normalisation of the weight matrix, W. Therefore, their results are expected to be very similar.

We now focus on the best-performing method ("LS" algorithm with "rbf" kernel and "BrownC" feature set) and explore its parameter space in more detail. In Figures 2–4, we plot the average performance scores for the 64 rumours as a function of the σ-parameter for different numbers of annotated messages. For comparison we also plot three benchmark models: a random classifier, which randomly assigns a message to a class with probability $1/3$, a weighted random classifier, which classifies a message in proportion to the class-frequency in a rumour, and the majority model, which assigns all messages to the majority class of a rumour.

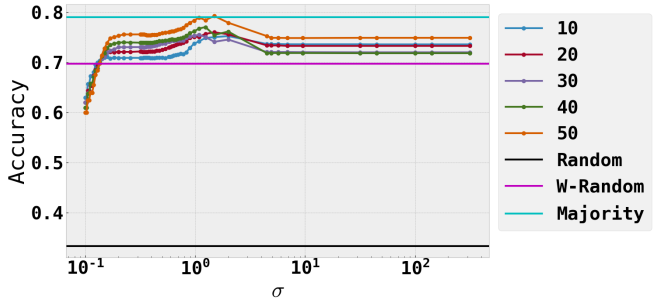


Fig. 2: Accuracy of the LS algorithm with rbf kernel and Brown cluster features against σ-parameter for several numbers of annotated messages N.

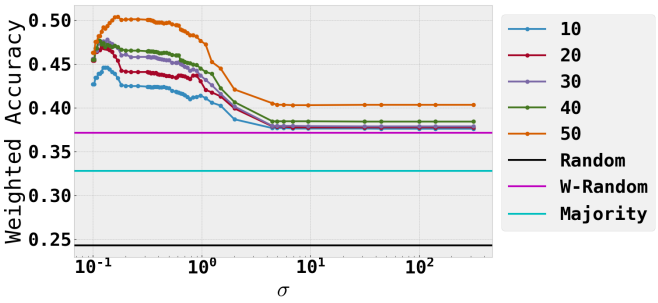


Fig. 3: Weighted accuracy of the LS algorithm with rbf kernel and Brown cluster features against σ-parameter for several numbers of annotated messages N.

We observe that the accuracy and weighted accuracy increase and the entropy decreases as the number of annotated messages increases, as expected. The more initial information the algorithm has, the better it performs. In addition, in most cases the models outperform the benchmark models [6].

In more detail, we observe that all metrics have a constant plateaux for $\sigma \gtrsim 5$. These values suppress the

6. The majority model outperforms the semi-supervised models on the accuracy score for some values of the parameters, but this is an artefact of the biased dataset, which becomes evident when looking at the weighted accuracy, Figure 3, and entropy, Figure 4.

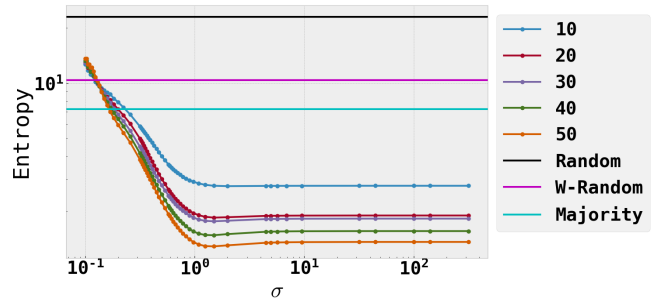


Fig. 4: Entropy of the LS algorithm with rbf kernel and Brown cluster features against σ-parameter for several numbers of annotated messages N. The y-axis is in log-scale to highlight the local minimum of σ.

Method	Max Accuracy	Max Weighted Accuracy	Max F1-Score	Accuracy at opt param	F1-Score at opt param
LP-BrownC-rbf	0.7822	0.4995	0.4606	0.7434	0.4500
LP-BrownC*-rbf	0.7604	0.4891	0.4341	0.7068	0.4151
LP-Ling-rbf	0.7529	0.4458	0.3997	0.6725	0.3829
LP-BrownC & Ling-rbf	0.7678	0.4736	0.4318	0.7083	0.4228
LP-Ngrams-rbf	0.7593	0.4389	0.3722	0.7001	0.3577
LP-BrownC-knn	0.7435	0.4112	0.3713	0.7141	0.3713
LS-BrownC-rbf	0.793	0.5037	0.4763	0.7489	0.4666

TABLE 1: Summary of performance scores for different methods. The first column contains the method (algorithm-feature set-kernel). The next four columns contain the maximum scores (occurring at different parameters) for $N = 50$ annotated messages. The last two columns contain the accuracy and F1-score at the optimal parameter, i.e. the parameter where the weighted accuracy is maximised.

exponent of the kernel (1), resulting in a similarity matrix (cf. eq. (1)) whose elements are all very close to 1. Therefore, all messages appear to be very similar to each other, hence giving the same prediction. For smaller values of σ, the messages become distinguishable in the graph representation, resulting in an increase of accuracy and weighted accuracy, where the entropy has a local minimum. For very small values of σ, the exponent in (1) becomes too large, hence, the elements of the similarity matrix become too small and the messages are very weakly connected in the graph representation, resulting in poor performance. It is the region at $\sigma \sim \mathcal{O}(1)$, where the accuracy scores have a local maximum and the entropy has a local minimum.

Focusing on the accuracy and entropy, the local optimal occurs at value $\sigma \sim 0.85$, whereas the weighted accuracy shows a fluctuating plateaux for $0.2 < \sigma \lesssim 1$. Combining the conclusions from the three metrics, we choose $\sigma = 0.85$ as the optimal value.

The remaining methods considered in Table 1 behave similarly, showing the same qualitative patterns, although the location of the optimal parameter may differ.

In Figure 5, we plot the distribution of the accuracies of the 64 rumours for the LS algorithm with rbf kernel and $\sigma = 0.85$. We notice that as more messages get annotated, the distribution is shifted to higher values. For $N = 50$, more than half of the rumours have accuracy greater than 80%. Only two rumours show an accuracy less than random, which will be investigated in future work, see Section 5. Some rumours have low accuracy because one or two classes are not present in the first 50 annotated messages. We aim to resolve such cases in future work, see also the discussion in Section 5.

Here, we selected σ so that it optimises the average performance scores. However, the messages in different rumours might have distinct spread in the feature space, hence, requiring a varying σ that depends on the particular rumour.

In [37], the authors proposed a heuristic method for determining the σ of individual datasets (here rumours). Particularly, they find the minimum spanning tree of labelled data, from which they estimate the minimum distance between two nodes that belong on different classes. Then σ is set to one third of that distance, following the rule of 3σ of the normal distribution.

In Figure 6, we plot the performance scores of the LS with tuned $\sigma = 0.85$ and the LS with σ dynamically determined using the heuristic of [37]. We observe that

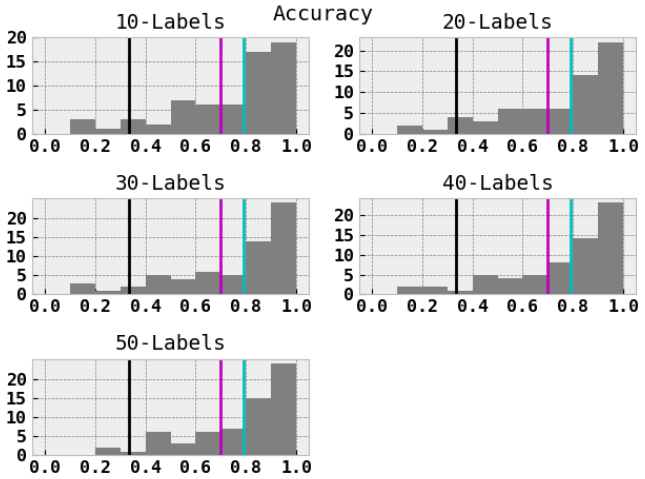


Fig. 5: Distribution of rumour accuracies for increasing number of annotated messages. The vertical lines indicate the accuracy of the benchmark models, random (black), weighted random (magenta) and majority (cyan).

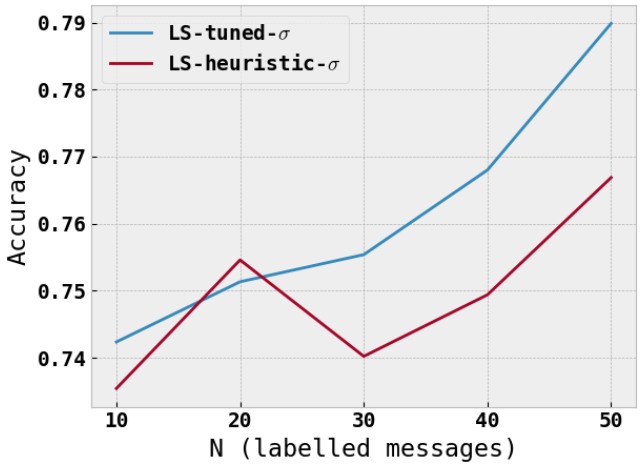
the accuracy of the "tuned" method is higher than that of the "heuristic" method; however, the latter outperforms the former in the weighted accuracy, indicating that the "heuristic" σ method is better for biased datasets.

Although the "heuristic" method underperforms in terms of accuracy, it is sometimes useful in a real-world system, which operates on corpus other than Twitter.

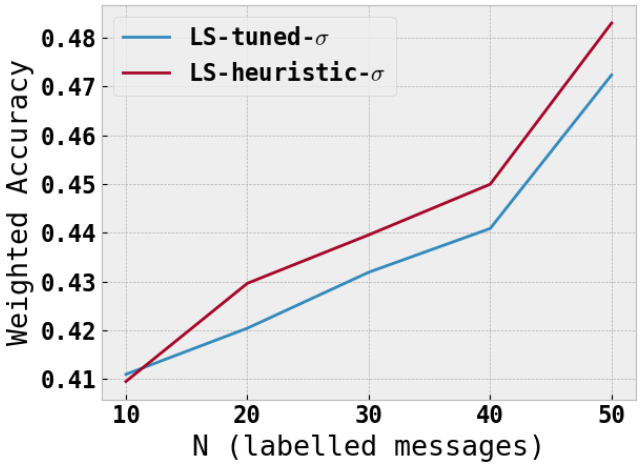
4.2 Validation and Comparison

We validate the proposed algorithm on two datasets in [16] and [17] and compare the performance of our approach with the Gaussian Processes of [16]. We choose to compare against this study for the following two reasons. First, its dataset and algorithms are publicly available. This allows us to make a direct comparison on the same dataset. Second, the authors of this work use a hybrid approach, which, to the best of our knowledge, is among the state-of-the-art in the academic literature.

The authors of [16] considered Gaussian Processes in three different training methods. The first method (here denoted as "GP") involves training only on the first N annotated messages from the rumour under consideration (the target rumour). In the second method ("GPPooled"), a GP model is trained on messages from other rumours in the



(a) Accuracy



(b) Weighted accuracy

Fig. 6: Average performance of the LS method with tuned and heuristic method for finding σ

dataset (the reference rumours) combined with the first N messages from the target rumour. The final configuration ("GPICM") is similar to the second one, but instead weighs the influence from the reference rumours.

We focus on the Brown clusters excluding the bag-of-words features. Our methods consist of the LS algorithm with rbf kernel and σ either tuned to $\sigma = 0.85$ or determined by the heuristic approach, described in the previous section, for each rumour.

4.2.1 London Riots Dataset

The dataset in [16] consists of seven rumours from the London riots in 2011. Due to anonymisation of the dataset, messages are replaced with their features, i.e. the 1,000 Brown clusters and bag of words. Therefore, we perform no pre-processing of the messages and work directly with their feature representation.

In [16], the authors trained a Gaussian Process (GP) [7] using only original tweets and validated it on a set that included both original tweets and retweets. Similarly, we annotate the first $N = \{10, 20, 30, 40, 50\}$ original tweets and compute the performance scores using all the remaining tweets. Here, we are not able to simply assign every retweet to the same class as its original tweet because the dataset has no retweet id information, from which one can associate retweets to original tweets. Instead, the dataset includes a tag identifying whether the message is a retweet or not. Therefore, the retweets participate in the algorithm as "original tweets".

The accuracy and weighted accuracy of the two proposed semi-supervised methods as a function of N are plotted in Figure 7. For comparison, we also plot the performance scores of the three GP methods and benchmark models. Regarding the LS algorithm, we observe that the performance scores increase as more tweets get annotated. Particularly, the tuned σ method achieves an accuracy of 83.2% and 84.9%, whereas the "heuristic" σ method scores 81.9% and 82.9%, at $N = 40$ and $N = 50$ respectively. All performance scores show that the LS method outperforms all other methods for $N \geq 40$. Particularly, it outperforms the "GP" method, which is actually a semi-supervised approach for Gaussian Processes, for all N. Although the remaining two GP methods achieve higher performance at early stages, they suffer from scalability and speed issues, hence, they are inefficient for quick message stance classification in a rapid-response live system.

For example, when applied on this dataset, consisting of 7 rumours and 7297 tweets (which is a moderate number, for real-life situations), the GP methods required about a week of training, on a 12-core machine [8]. Given that frequent retraining would be required for any live system, this demonstrates that supervised methods, though accurate, cannot scale up, therefore limiting their usefulness for realistic systems.

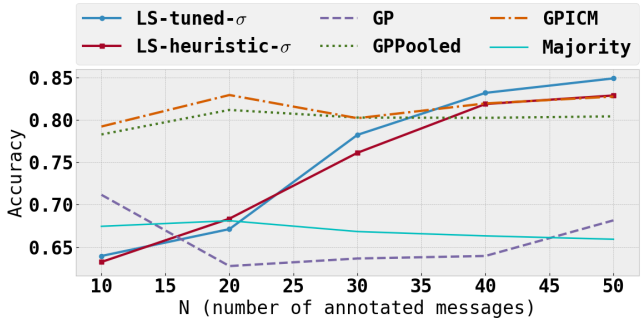
4.2.2 PHEME Dataset

We also compare the two methods on another publicly available dataset [17]. This set consists of tweet conversations, collected in association with 9 breaking news stories. The conversations are organised in threads the root of which is the initiating rumour tweet, accompanied by the corresponding replies. The tweets are annotated for support, certainty and evidentiality. In order to align this dataset with the purpose of this study, we group threads by rumour. In our nomenclature there are two levels of support in this set; whether the initial tweet supports or not the rumour and whether the subsequent tweets support the initial tweet's claim. We straighten this two-step relation, by resolving the support of each tweet against the rumour, and update the annotation accordingly. For example if the initial tweet supports the rumour claim it is annotated as such. If a subsequent tweet (reply) negates the initial tweet with certainty, then it is annotated as not supporting the rumour claim.

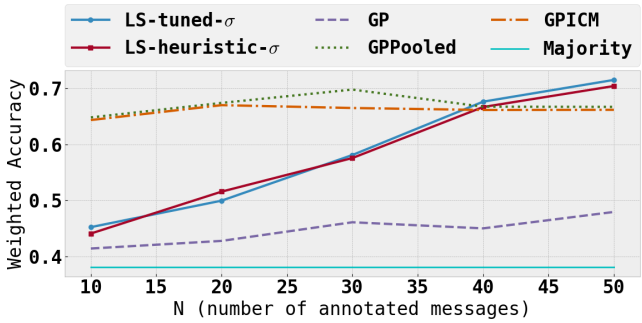
The dataset contains 297 threads containing 4561 tweets (including retweets), spanning 138 rumours organised in 9 stories. For the purpose of this study, as explained in previous sections, we select only the rumours containing

7. For an introductory review on GP see [40].

8. Dual Intel(R) Xeon(R) CPU E5-2630 v2 @ 2.60GHz, 256GB RAM.

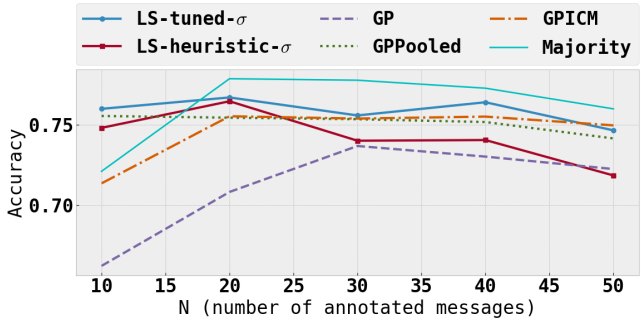


(a) Accuracy

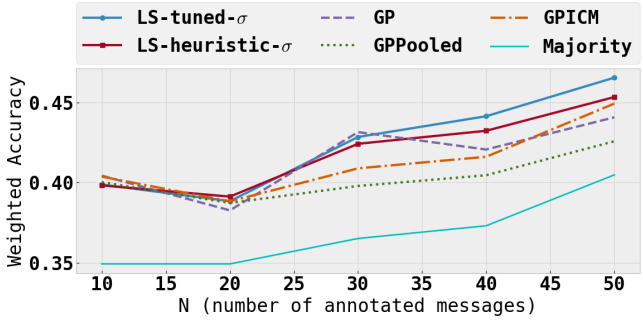


(b) Weighted accuracy

Fig. 7: Average performance scores of the LS (solid line with round markers), the three GP methods (non-solid lines) and benchmark models (solid lines) of the 7 London riots rumours.



(a) Accuracy



(b) Weighted accuracy

Fig. 8: Average performance scores of the LS (solid line with round markers), the three GP methods (non-solid lines) and benchmark models (solid lines) of the 23 PHEME rumours.

at least 50 English original tweets. The final number of rumours we are using from this dataset is therefore 23 containing 2233 (original English) tweets.

The accuracy and weighted accuracy of the models are presented in Figure 8. Looking at the accuracy of the majority model in Figure 8a, we conclude that the dataset is strongly biased, with most messages belonging in a single class. Therefore, we focus on the weighted accuracy, which suppress the majority model as well as the "GPPooled" model. Both versions of the semi-supervised LS methods, score at least as well as the GPs from the very early stages of the rumours' development and outperform the GP methods for $N \geq 40$ labelled messages.

4.2.3 Remarks

Following the conclusions in [16], we make the following observations.

- The performance of LS increases as more tweets get annotated. This behaviour is expected because a semi-supervised algorithm relies on limited information, the more the better.
- "GP" resembles the semi-supervised learning, as only a limited number of tweets from the target rumour are used for training. Comparing to the results presented here, we achieve at least a similar accuracy from early on, $N = 10$, on both datasets, however, the performance of the LS methods increases with N, exceeding 80% at $N = 40$ on the "London Riots" dataset.
- The weighted accuracy of the proposed models at $N \geq 40$ exceeds the accuracy of all three methods in [16] on both data-sets.
- "GPICM" in [16] outperforms the LS method for $N < 30$ on the "London Riots" dataset (but not on the "Pheme" dataset). However, this might be an artefact due to the lack of messages' diversity in the "London Riots" dataset. GP was trained only on messages about a particular topic, the London riots, hence, all messages in the pooled rumours are relevant to the messages in the target rumour, achieving a better score due to over-fitting. This might be the reason why the GP methods do not perform significantly better at low N on the second dataset, where messages from diverse topics exist.
- "GPICM" and "GPPooled" are particularly inefficient in a live system where both speed and accuracy are essential. Training a new model as new messages arrive slows down the process, particularly, when supervised training is performed on a large dataset, as in "GPPooled" and "GPICM".
- Finally, it should be noted that the "GP" method with no tweets from the rumour under consideration ("Leave-one-out"), simulates the situation when a completely unknown rumour is examined. Such a design could address our concerns about scalability if it performed well enough, since one would only

need to train a model once. This set-up was examined but consistently underperformed every other result reported here, which is why it is not included in the plots.

Overall, the proposed algorithm achieves a better performance and is much faster than a GP (as considered in [16]). The LS scales as $\mathcal{O}(n^2)$. Specifically, the times required to process the rumours in our dataset fit the polynomial time$(n) = 2.06 \cdot 10^{-7} n^2 + 4.47 \cdot 10^{-5} n - 9.32 \cdot 10^{-3}$ seconds, i.e. a rumour with 1,000 messages is processed in 0.24 seconds [9].

In the previous comparison we have focused on the accuracy (measured with three different metrics) of each algorithm. Here, we would like to emphasize another point of comparison, namely scaling. The top-performing Gaussian Process algorithms, i.e. "GPPooled" and "GPICM" rely on a sizeable reference library of messages, over which training is performed. In a real-life system, dealing with millions of messages and hundreds of wildly diverse rumours, this reliance cripples performance. One would first have to train on this reference library, and then apply the resulting model on arriving messages. Moreover, as the messages that do not belong to the reference library grow in number, periodically retraining will become necessary, now on an even larger library. In other words, when one considers the complexity of Gaussian Process algorithms, which is $\mathcal{O}(n^3)$ or at best $\mathcal{O}(n^2)$ [40], one needs to remember that, in these cases, n refers to the number of reference messages. Contrary to that, "LS", which scales as $\mathcal{O}(n^2)$, only involves the messages of the rumour under investigation and is completely agnostic to other rumours, thus n is a significantly smaller number. Furthermore, since each incoming rumour is treated independently there is no training stage and no need for retraining. It therefore becomes clear, that "LS" is significantly better at performing in realistic environments. Practically, as already mentioned, the GP methods when applied on the PHEME data set, consisting of 2233 messages (which is a moderate number, for real-life situations) required almost 14 days of training, on a 12-core machine. Given that frequent retraining would be required for any live system, this demonstrates that supervised methods, though accurate, cannot scale up, therefore limiting their usefulness for realistic systems. In contrast, the LS method would take 1.1 seconds for a rumour of size 2233.

5 Conclusion and Outlook

In the modern world dominated by social media interactions, unverified stories can spread quickly, having a huge impact on people's life, particularly on situations of crisis, such as terrorist attacks, natural disaster, accidents, or even a financial impact. Determining the trustworthiness of information is a challenging and open problem. Particularly, on situations of crisis, the veracity of rumours must be resolved as quickly as possible. Therefore, speed and classification performance are equally important. Several methods have been proposed to automate the identification of rumour veracity. Towards this goal, the classification of messages, i.e. whether they support, deny or question a rumour, is a crucial feature [10], as people tend to correctly judge a situation collectively ("wisdom of the crowd"). However, this task often requires manual effort. The aim is to automate this process as much as possible and reduce the burden on end-users, i.e. a fast process that requires minimum input information from an end-user, classifies whether the messages support, deny or are neutral to the rumour and feeds the classification to the rumour veracity assessor.

Having reviewed the literature, we found that most methods for message stance classification rely on supervised machine learning [15]. We argued why such algorithms do not address the problem satisfactorily, in terms of accuracy, computational speed and scalability, and instead we propose that existing semi-supervised algorithms tackle the problem more efficiently, especially when dealing with large and diverse datasets. We focus on a family of graph-based semi-supervised algorithms, the Label Propagation and Label Spreading. The algorithms' accuracy increases as more messages get annotated. In a real scenario, a software tool, with a user-tailored interface, can display the first tens of messages, which are able to be annotated by the end-user in a very short time. Our study shows that the proposed algorithms are fast and accurate, exceeding 80% average accuracy.

We compared to Gaussian processes used in a supervised and semi-supervised setting. The results show that the graph-based algorithms are faster and at least as accurate, particularly as more annotated messages become available, therefore they are more effective for implementation in a rapid-response software system.

Despite their success, these algorithms face a few challenges, some of which have been addressed, while others require further improvements, which will be explored in future work. We briefly mention these issues below.

First, from a usability point of view, the semi-supervised method proposed in this work requires a set of annotated messages. In a live-system, an analyst or end-user might urgently need an estimate of the message stance and hence of the rumour veracity. For such scenarios, we have developed a supervised logistic regression model, trained on a subset of our dataset, which can be applied to the new messages. This model captures average message characteristics, which are unrelated to the specific rumour, such as, messages that have the word "believe" without negation are more likely to support a statement. This is a simple solution that address the "cold-start" problem when no annotated messages are available. Other supervised models available in the literature could be equally applied, hence integrating multiple approaches into one tool. We intend to address this issue systematically in future work.

Another solution to this problem could be online learning algorithms [41], [42], which aim to update a model as sequences of data become available and are faster and more efficient than batch-learning supervised algorithms. However, such an approach has not been developed within the context of message stance classification, hence a complete study, end-to-end implementation and its comparison to semi-supervised methods are left for future work.

A second issue regarding the the LP and LS is that the number of classes is implied by the annotated messages.

9. On a laptop with 16GB RAM and Intel(R) Core i7-3610QM CPU @ 2.30GHz

For example, if the first N messages belong only to two classes, then all other messages will be classified into one of these two classes. In future work, we aim to improve the algorithm, so that if a message is too distinct from the annotated ones, then it gets classified into a new cluster.

ACKNOWLEDGEMENTS

This work was partly supported by the UK Defence Science and Technology Laboratory under Centre for Defence Enterprise grant DSTLX-1000107083. We thank Colin Singleton, Chris Willis and Nicholas Walton for their helpful comments during the project. We would also like to thank Dr. Matthew Edgington and Alan Pilgrim for their assistance in annotating part of the dataset.

REFERENCES

[1] The Wall Street Journal, "Hurricane Harvey Victims Turn to Social Media for Assistance," 2017, https://www.wsj.com/articles/hurricane-harvey-victims-turn-to-social-media-for-assistance-1503999001 (Accessed on 15-Dec-2017).

[2] MTV, "The People Of Manchester Are Using Social Media To Help Each Other Following Explosion At Ariana Grande Concert," 2017, http://www.mtv.co.uk/ariana-grande/news/people-manchester-using-social-media-to-help-each-other-following-terrorist-attack (Accessed on 15-Dec-2017).

[3] B. Future, "Lies, propaganda and fake news: A challenge for our age," 2017, http://www.bbc.com/future/story/20170301-lies-propaganda-and-fake-news-a-grand-challenge-of-our-age (Accessed on 15-Dec-2017).

[4] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on twitter," in *Proceedings of the 20th international conference on World wide web*. ACM, 2011, pp. 675–684.

[5] M. R. Morris, S. Counts, A. Roseway, A. Hoff, and J. Schwarz, "Tweeting is believing?: understanding microblog credibility perceptions," in *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*. ACM, 2012, pp. 441–450.

[6] C. Castillo, M. Mendoza, and B. Poblete, "Predicting information credibility in time-sensitive social media," *Internet Research*, vol. 23, no. 5, pp. 560–588, 2013.

[7] A. Gupta, H. Lamba, P. Kumaraguru, and A. Joshi, "Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy," in *Proceedings of the 22nd international conference on World Wide Web*. ACM, 2013, pp. 729–736.

[8] J. R. C. Nurse, I. Agrafiotis, M. Goldsmith, S. Creese, and K. Lamberts, "Two sides of the coin: measuring and communicating the trustworthiness of online information," *Journal of Trust Management*, vol. 1, no. 1, p. 5, 2014.

[9] M. Alrubaian, M. Al-Qurishi, M. Hassan, and A. Alamri, "A credibility analysis system for assessing information on twitter," *IEEE Transactions on Dependable and Secure Computing*, 2016.

[10] G. Giasemidis, C. Singleton, I. Agrafiotis, J. R. C. Nurse, A. Pilgrim, C. Willis, and D. V. Greetham, "Determining the veracity of rumours on twitter," in *Social Informatics: 8th International Conference, SocInfo 2016, Bellevue, WA, USA, November 11-14, 2016, Proceedings, Part I*, E. Spiro and Y.-Y. Ahn, Eds. Cham: Springer International Publishing, 2016, pp. 185–205. [Online]. Available: https://doi.org/10.1007/978-3-319-47880-7_12

[11] M. Mendoza, B. Poblete, and C. Castillo, "Twitter under crisis: Can we trust what we rt?" in *Proceedings of the first workshop on social media analytics*. ACM, 2010, pp. 71–79.

[12] S. Vosoughi, "Automatic detection and verification of rumors on twitter," Ph.D. dissertation, Massachusetts Institute of Technology, 2015.

[13] K. Wu, S. Yang, and K. Q. Zhu, "False rumors detection on sina weibo by propagation structures," in *Data Engineering (ICDE), 2015 IEEE 31st International Conference on*. IEEE, 2015, pp. 651–662.

[14] Z. Zhao, P. Resnick, and Q. Mei, "Enquiring minds: Early detection of rumors in social media from enquiry posts," in *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2015, pp. 1395–1405.

[15] A. Zubiaga, A. Aker, K. Bontcheva, M. Liakata, and R. Procter, "Detection and resolution of rumours in social media: A survey," *ACM Comput. Surv.*, vol. 51, no. 2, pp. 32:1–32:36, Feb. 2018. [Online]. Available: http://doi.acm.org/10.1145/3161603

[16] M. Lukasik, T. Cohn, and K. Bontcheva, "Classifying tweet level judgements of rumours in social media," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2015, pp. 2590–2595. [Online]. Available: http://www.aclweb.org/anthology/D15-1311

[17] A. Zubiaga, M. Liakata, R. Procter, G. Wong Sak Hoi, and P. Tolmie, "Pheme rumour scheme dataset: journalism use case," Apr 2016. [Online]. Available: https://figshare.com/articles/PHEME_rumour_scheme_dataset_journalism_use_case/2068650/2

[18] R. Procter, F. Vis, and A. Voss, "Reading the riots on twitter: methodological innovation for the analysis of big data," *International journal of social research methodology*, vol. 16, no. 3, pp. 197–214, 2013.

[19] C. Andrews, E. Fichet, Y. Ding, E. S. Spiro, and K. Starbird, "Keeping up with the tweet-dashians: The impact of'official'accounts on online rumoring," in *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*. ACM, 2016, pp. 452–465.

[20] V. Qazvinian, E. Rosengren, D. R. Radev, and Q. Mei, "Rumor has it: Identifying misinformation in microblogs," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 1589–1599.

[21] S. Hamidian and M. T. Diab, "Rumor identification and belief investigation on twitter." in *WASSA@ NAACL-HLT*, 2016, pp. 3–8.

[22] S. M. Mohammad, P. Sobhani, and S. Kiritchenko, "Stance and sentiment in tweets," *ACM Transactions on Internet Technology (TOIT)*, vol. 17, no. 3, p. 26, 2017.

[23] G. J. Werner, V. Prabhakaran, M. Diab, and O. Rambow, "Committed belief tagging on the factbank and lu corpora: A comparative study," in *Proceedings of the Second Workshop on Extra-Propositional Aspects of Meaning in Computational Semantics (ExProM 2015)*, 2015, pp. 32–40.

[24] S. Hamidian and M. T. Diab, "Rumor detection and classification for twitter data," in *Proceedings of the Fifth International Conference on Social Media Technologies, Communication, and Informatics (SOTICS)*, 2015, pp. 71–77.

[25] L. Zeng, K. Starbird, and E. S. Spiro, "# Unconfirmed: Classifying Rumor Stance in Crisis-Related Social Media Messages," in *Tenth International AAAI Conference on Web and Social Media*, 2016.

[26] M. Lukasik, K. Bontcheva, T. Cohn, A. Zubiaga, M. Liakata, and R. Procter, "Using gaussian processes for rumour stance classification in social media," *arXiv preprint arXiv:1609.01962*, 2016.

[27] M. Lukasik, T. Cohn, and K. Bontcheva, "Classifying tweet level judgements of rumours in social media," *arXiv preprint arXiv:1506.00468*, 2015.

[28] Z. Jin, J. Cao, Y. Zhang, and J. Luo, "News verification by exploiting conflicting social viewpoints in microblogs." in *AAAI*, 2016, pp. 2972–2978.

[29] A. Zubiaga, E. Kochkina, M. Liakata, R. Procter, and M. Lukasik, "Stance classification in rumours as a sequential task exploiting the tree structure of social media conversations," *arXiv preprint arXiv:1609.09028*, 2016.

[30] E. Kochkina, M. Liakata, and I. Augenstein, "Turing at semeval-2017 task 8: Sequential approach to rumour stance classification with branch-lstm," *arXiv preprint arXiv:1704.07221*, 2017.

[31] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[32] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *ACM SIGKDD Explorations Newsletter*, vol. 19, no. 1, pp. 22–36, 2017.

[33] W. Ferreira and A. Vlachos, "Emergent: a novel data-set for stance classification," in *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies*, 2016, pp. 1163–1168.

[34] O. Owoputi, B. O'Connor, C. Dyer, K. Gimpel, N. Schneider, and N. A. Smith, "Improved part-of-speech tagging for online conversational text with word clusters," in *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14,*

2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA, 2013, pp. 380–390. [Online]. Available: http://aclweb.org/anthology/N/N13/N13-1039.pdf

[35] J. Yi, T. Nasukawa, R. Bunescu, and W. Niblack, "Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques," in *null*. IEEE, 2003, p. 427.

[36] O. Chapelle, B. Schlkopf, and A. Zien, *Semi-Supervised Learning*, 1st ed. The MIT Press, 2010.

[37] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," Carnegie Mellon University, Tech. Rep., 2002.

[38] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Advances in Neural Information Processing Systems 16*, S. Thrun, L. K. Saul, and B. Schölkopf, Eds. MIT Press, 2004, pp. 321–328. [Online]. Available: http://papers.nips.cc/paper/2506-learning-with-local-and-global-consistency.pdf

[39] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[40] S. Roberts, M. Osborne, M. Ebden, S. Reece, N. Gibson, and S. Aigrain, "Gaussian processes for time-series modelling," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 371, no. 1984, 2013. [Online]. Available: http://rsta.royalsocietypublishing.org/content/371/1984/20110550

[41] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*, 1st ed. Cambridge University Press, 2014.

[42] A. Rakhlin and K. Sridharan, "A tutorial on online supervised learning with applications to node classification in social networks," 2016. [Online]. Available: http://arxiv.org/abs/1608.09014

Georgios Giasemidis received his DPhil (PhD) in theoretical physics from University of Oxford in 2013. Since then, he is a senior analyst and data scientist at CountingLab LTD, a spin-out from the Center for the Mathematics of Human Behaviour, University of Reading. His research interests and experience include big data analytics, complex and social networks, electricity demand forecasting, low-voltage networks, image-processing and machine learning algorithms for classification and clustering of big data. In 2014, he won the third award at the Global Energy Forecasting Competition.

Nikolaos Kaplis is a Data Scientist at CountingLab, a spin-out from the Center for the Mathematics of Human Behaviour, University of Reading. He obtained his DPhil from the University of Oxford (Magdalen) in 2013. He studied Theoretical Physics and Applied Mathematics. He works on big-data analytics, machine learning, networks and AI.

Ioannis Agrafiotis is a research fellow (senior researcher) in the Department of Computer Science at the University of Oxford, where he currently explores novel ways to capture organisational cyber harm and risk. He is also working on a project aiming at detecting insider threats. His research interests include automated network defence and business process modelling, information trustworthiness, online privacy and dynamic consent, insider threat and anomaly detection.

Jason R.C. Nurse is a Lecturer in Cyber Security in the School of Computing at the University of Kent. Prior to this he was a Senior Research Fellow in the Department of Computer Science at the University of Oxford and a fellow at Wolfson College, Oxford. His research interests include the information provenance and trust, social media studies, human factors of security, and services security. Nurse received his PhD from the University of Warwick in 2010 in the topic of Internet security for corporations. In 2014, he was selected as a Rising Star for his research into cyber security and privacy, as a part of the EPSRCs Recognising Inspirational Scientists and Engineers (RISE) awards campaign.